

In the Claims

1. (Previously Presented) A method for processing a header portion of a message, comprising:

establishing a legacy protocol, wherein said legacy protocol defines at least one legacy parameter for a header portion of a message, and wherein said legacy protocol defines a fixed legacy header length;

receiving an inbound message having a header portion;

allocating a memory portion from the computer memory, said memory portion having a depth corresponding to said fixed legacy header length;

pushing said header portion of said inbound message onto said memory portion thereby forming a received header, wherein the header portion is pushed onto said memory portion such that said header portion is truncated to form the received header when a length of said header portion is greater than said depth of said memory portion corresponding to said fixed legacy header length and wherein said header portion is not truncated when a length of said header portion is not greater than said depth of said memory portion, such truncation causing any header parameters associated with an upgraded protocol to be removed from said header portion; and

processing said received header according to said legacy protocol.

2. (Previously Presented) The method according to claim 1, further comprising:
establishing said upgraded protocol, wherein said upgraded protocol includes said at least one legacy parameter of said legacy protocol, wherein said upgraded protocol defines at least one upgraded header parameter for said header portion, and wherein said upgraded protocol defines a fixed upgraded header length;

wherein said memory portion has a depth corresponding to said upgraded header length;

wherein said received header of said inbound message is processed according to said upgraded protocol if at least one upgraded header parameter is pushed on the memory portion; and

wherein said received header of said inbound message is processed according to said legacy protocol when no upgraded header parameters are pushed on the memory portion.

3. (Previously Presented) The method according to claim 2 further comprising padding said memory portion with default padding values when said header portion of said inbound message does not fill said memory portion.

4. (Previously Presented) The method according to claim 1, wherein said legacy parameter comprises a value-type pair.

5. (Previously Presented) The method according to claim 1, wherein said inbound message includes a data portion and wherein said header portion is pushed onto said memory portion after said data portion.

6. (Canceled)

7. (Canceled)

8. (Canceled)

9. (Canceled)

10. (Canceled)

11. (Previously Presented) A method for processing a header portion of a message, comprising:

establishing a legacy protocol, wherein said legacy protocol defines at least one legacy parameter for a header portion of a message, and wherein said legacy protocol defines a fixed legacy header length;

receiving an inbound message having a header portion;

allocating a memory portion from the computer memory, said memory portion having a depth corresponding to said fixed legacy header length;

pushing said header portion of said inbound message onto said memory portion thereby forming a received header, wherein the header portion is pushed onto said memory portion such that said header portion is truncated to form the received header when a length of said header portion is greater than said depth of said memory portion corresponding to said fixed legacy header length and wherein said header portion is not truncated when a length of said header portion is not greater than said depth of said memory portion, such truncation causing any header parameters associated with an upgraded protocol to be removed from said header portion;

processing said received header according to said legacy protocol;

constructing a legacy header according to said legacy protocol;

appending said legacy header to outbound data thereby creating an outbound message; and

sending said outbound message.

12. (Previously Presented) The method according to claim 11, further comprising:

establishing said upgraded protocol, wherein said upgraded protocol includes said at least one legacy parameter of said legacy protocol, wherein said upgraded protocol defines at least one upgraded header parameter, and wherein said upgraded protocol defines a fixed upgraded header length;

wherein said memory portion has a depth corresponding to said upgraded header length;

wherein said received header of said inbound message is processed according to said upgraded protocol if at least one upgraded header parameter is pushed on the memory stack;

wherein said received header of said inbound message is processed according to said legacy protocol if no upgraded header parameters are pushed on the memory stack;

constructing an upgraded header according to said upgraded protocol; and

appending said upgraded header to outbound data.

13. (Previously Presented) The method according to claim 12 further comprising pushing said legacy parameter onto said memory portion before said upgraded parameter is pushed onto said memory portion.

14. (Previously Presented) The method according to claim 11 further comprising:
receiving said inbound message from an upper layer application having a header portion in an upper layer format; and

sending said outbound message to a lower layer application.

15. (Previously Presented) Software for processing at least one inbound message, the software embodied on a computer-readable medium and operable, when executed by a computer, to:

establish a legacy protocol, wherein said legacy protocol defines at least one legacy parameter for a header portion of a message, and wherein said legacy protocol defines a fixed legacy header length;

allocate a memory portion from the computer memory, said memory portion having a depth corresponding to said fixed legacy header length;

push said header portion of said inbound message onto said memory portion wherein said data portion is pushed onto said stack first, wherein the header portion is pushed onto said memory portion such that said header portion is truncated to form the received header if when a length of said header portion is greater than said depth of said memory portion corresponding to said fixed legacy header length and wherein said header portion is not truncated when a length of said header portion is not greater than said depth of said memory portion, such truncation causing any header parameters associated with an upgraded protocol to be removed from said header portion; and

process said received header according to said legacy protocol.

16. (Previously Presented) The software according to claim 15, further operable to:

establish said upgraded protocol, wherein said upgraded protocol includes said at least one legacy parameters of said legacy protocol, wherein said upgraded protocol defines at least one upgraded header parameter, and wherein said upgraded protocol defines a fixed upgraded header length;

wherein said memory portion has a depth corresponding to said upgraded header length;

wherein said portion of said message is processed according to said upgraded protocol if at least one upgraded header parameter is pushed onto said memory portion; and

wherein said portion of said message is processed according to said legacy protocol if no upgraded header parameters are pushed onto said memory portion.

17. (Previously Presented) The software according to claim 16 further operable to pad said memory portion with default padding values when said message does not fill said memory portion.

18. (Canceled)

19. (Canceled)

20. (Previously Presented) Software for processing inbound and outbound messages, the software embodied on a computer-readable medium and operable, when executed by a computer, to:

- establish a legacy protocol, wherein said legacy protocol defines at least one legacy parameter for a header portion of inbound and outbound messages, and wherein said legacy protocol defines a fixed legacy message length;

- receive an inbound message having a data portion;

- allocate a memory stack from said computer memory, said memory stack having a depth corresponding to said fixed legacy message length;

- push said inbound message onto said memory stack thereby forming at least a portion of said inbound message, wherein said data portion is pushed onto said stack proximate to a bottom of said stack, also wherein the header portion is pushed onto said memory portion such that said header portion is truncated to form the received header when a length of said header portion is greater than said depth of said memory portion corresponding to said fixed legacy header length and wherein said header portion is not truncated when a length of said header portion is not greater than said depth of said memory portion, such truncation causing any header parameters associated with an upgraded protocol to be removed from said header portion;

- process said portion of said inbound message according to said legacy protocol;

- construct a legacy header according to said legacy protocol;

- append said legacy header to outbound data thereby creating an outbound message of said fixed legacy message length; and

- send said outbound message.

21. (Previously Presented) The software according to claim 20, further operable to:

establish said upgraded protocol, wherein said upgraded protocol includes said at least one legacy parameter of said legacy protocol, wherein said upgraded protocol defines at least one upgraded header parameter, and wherein said upgraded protocol defines a fixed upgraded message length;

wherein said memory stack has a depth corresponding to said upgraded message length;

wherein said portion of said inbound message is processed according to said upgraded protocol if at least one upgraded header parameter is pushed onto said memory stack;

wherein said portion of said inbound message is processed according to said legacy protocol if no upgraded header parameters are pushed onto said memory stack;

construct an upgraded header according to said upgraded protocol; and

append said upgraded header to outbound data, wherein said at least one legacy parameter is proximate to said outbound data, thereby creating an outbound message of said fixed upgraded message length.

22. (Previously Presented) The software according to claim 21 further operable to:

receive said inbound message from an upper layer application having a data portion in an upper layer format; and

send said outbound message to a lower layer application.